

What is Walk-Forward Optimisation and why do we need it?

Walk forward optimisations (aka WFO) are essentially multiple overlapping in and out-of-sample optimisations.

For those of you who have forgotten what are in and out-of-sample optimisations, here is a refresher: <https://algotrading101.com/courses/13378/lectures/640556>

1 year		0.5 year					
In-sample		Out-sample					
		In-sample		Out-sample			
				In-sample		Out-sample	
						In-sample	
						Out-sample	
2015.06.30	2015.12.31	2016.06.30	2016.12.31	2017.06.30	2017.12.31	2018.06.30	

The above chart gives us a visual look at an WFO. In this case, the WFO consists of 5 sets of in and out-of-sample optimisations.

What are interested in are the performances of the 5 out-of-sample backtests (those in dark blue).

How to run Walk Forward Optimisations (WFO)

We shall use the same WFO chart for a case study.

1 year		0.5 year					
In-sample		Out-sample					
		In-sample		Out-sample			
				In-sample		Out-sample	
						In-sample	
						Out-sample	
2015.06.30	2015.12.31	2016.06.30	2016.12.31	2017.06.30	2017.12.31	2018.06.30	

The above WFO chart runs from 2015.07.01 to 2018.12.31 with 1 year in-sample periods with 0.5 years out-of-sample periods.

Here are the steps to run a walk forward optimisation:

1. Get all relevant data
2. Break data into multiple pieces
3. Run an optimisation to find the best parameters on the first piece of data (first in-sample)
4. Apply those parameters on the second piece of data (first out-of-sample)
5. Run an optimisation to find the best parameters on the next in-sample data
6. Apply those parameters on the next out-of-sample data
7. Repeat until you've covered all the pieces of data
8. Collate the performances of all the out-of-sample data

Step 1: Get all relevant data

The basic set of data you need is the price data of the financial product you are trading.

For example, if you are running a strategy that involves Uber and Lyft stocks, you need the prices of Uber and Lyft stocks over a period of time.

If your strategy requires any other data, which it normally does, you need those data as well.

These additional data could be the search traffic numbers of Uber and Lyft, amount of comments on ride-share drivers forums or S&P Index price etc.

Step 2: Break data into multiple pieces

Let's say that we are running our walk forward optimisation from 2015.07.01 to 2018.12.31. (Okay, I know Uber and Lyft weren't listed during 2015, let's just pretend they were.)

We break our data up into 10 pieces as seen in figure 1.

Our first piece of data is from 2015.07.01 to 2015.12.31. This is our first in-sample data.

Step 3: Run an optimisation to find the best parameters on the first piece of data

We will run an optimisation on that first in-sample data to find what strategy parameters were the best for that period.

In this example, we shall assume that our strategy is to buy Uber and short Lyft if there are more positive comments about Uber than Lyft in the ride-share drivers forums, vice versa. (Note that this is a grossly simplified strategy.)

Thus, a strategy parameter could be the number of positive comments for Uber divided by the number of positive comments for Lyft.

We shall call this PCUber/PCLyft.

To keep it simple, we have only one strategy parameter. However, most strategies usually have more than one parameter. (Note that having many strategy parameters does not necessarily make your strategy better!)

In our optimisation, we are trying to find out what value of PCUber/PCLyft should we fire our trades.

Let's assume that the optimisation results are out – we buy Uber and short Lyft when PCUber/PCLyft is 2 or higher and we short Uber and buy Lyft when PCUber/PCLyft is 0.5 or lower.

These parameters are known as the optimised parameters.

Step 4: Apply those parameters on the second piece of data

Our second piece of data is from 2016.07.01 to 2016.06.30. This is our first out-of-sample data.

We apply the optimised parameter values of 2 and 0.5 (for PCUber/PCLyft) to this second piece of data and check the performance of our strategy.

This performance is what we are interested in as it tells us if the data in the one year (in-sample) period has any predictive value in the proceeding half year (out-of-sample) period.

Step 5: Run an optimisation to find the best parameters on the next in-sample data

We repeat step 3 for our second in-sample data and find the best parameters for PCUber/PCLyft.

Our second in-sample data is from 2016.01.01 to 2016.12.31.

Step 6: Apply those parameters on the next out-of-sample data

We repeat step 4 for our second out-of-sample data.

Our second out-of-sample data is from 2017.01.01 to 2017.06.30.

Step 7: Repeat until you've covered all the pieces of data

We repeat this process until we reach our last piece of out-of-sample data.

The last piece of out-of-sample data is from 2018.07.01 to 2018.12.31.

Step 8: Collate the performances of all the out-of-sample data

Collate the performance of your strategy in all the out-of-sample data.

This would be the performance of your strategy using a walk forward optimisation.

This process is much more reliable than using a simple backtest or optimisation.

Why should we run Walk Forward Optimisations?

We run walk forward optimisations to reduce curve fitting (AKA overfitting) in our backtests and optimisations.

Check out "Chapter 21 > Buff Up Your Robot Responsibly - Optimisation Without Curve Fitting" if you have forgotten what's curve fitting.

A walk forward optimisation forces us to verify that we are adjusting our strategy parameters to signals (and not noise) in the past by constantly testing our optimised parameters in out-of-sample data.

Inexperience traders tend to spend a lot of time optimising every parameter on the entire set of past data. They then proceed to trade on these "optimised" parameters without running them on any out-of-sample data.

This sort of curve fitting is usually a recipe for disaster.

Another benefit is that we can see how our strategy performs as we change our parameters regularly to adapt to the market.